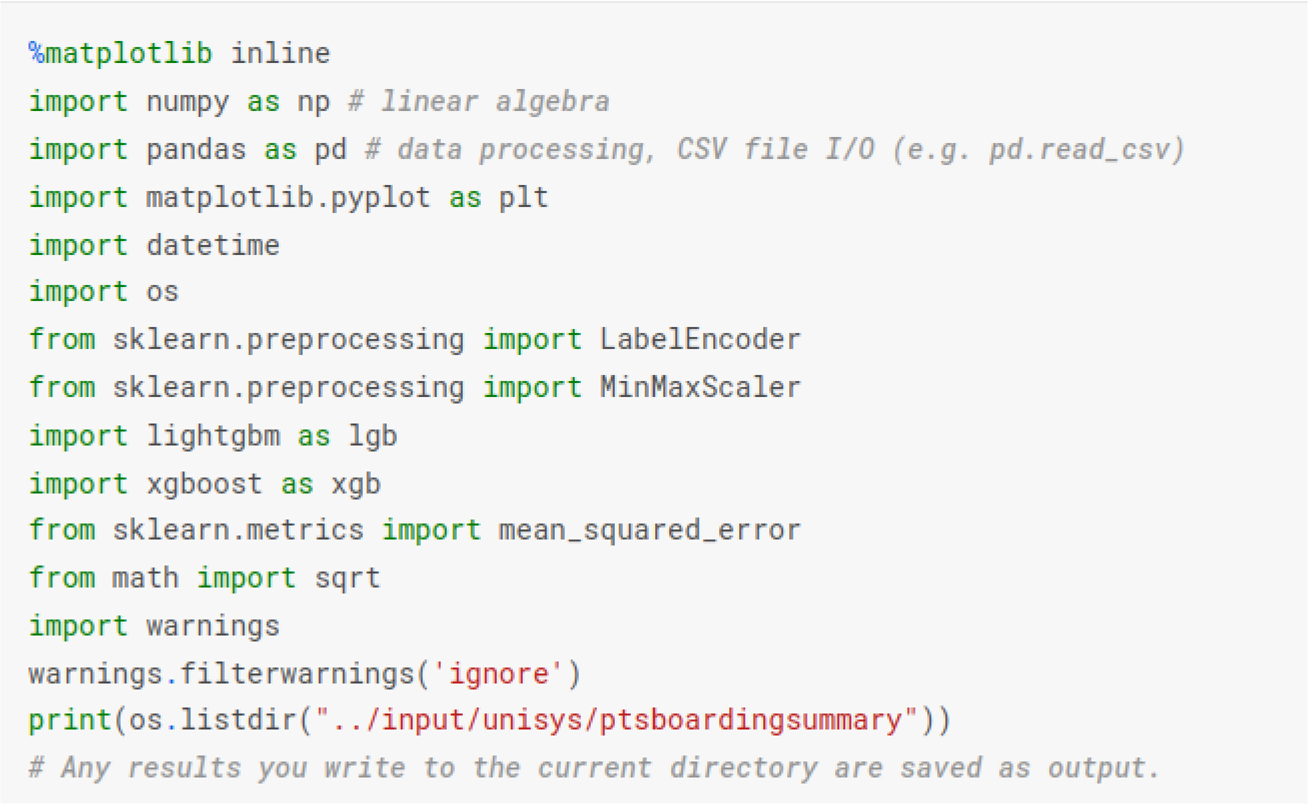
Phase 4: Development Part 2

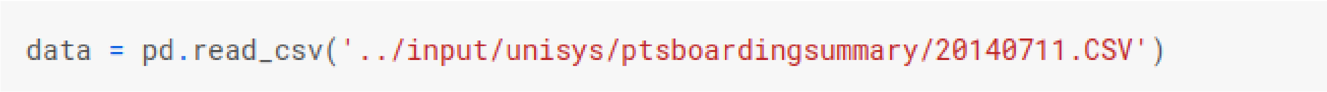
Problem: Public Transport Efficiency Analysis

1. The first step is to import necessary libraries (Pandas, NumPy, Matplotlib, and Seaborn) for data analysis and visualization in Python.



```
%matplotlib inline
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import datetime
import os
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
import lightgbm as lgb
import xgboost as xgb
from sklearn.metrics import mean_squared_error
from math import sqrt
import warnings
warnings.filterwarnings('ignore')
print(os.listdir("../input/unisys/ptsboardingsummary"))
# Any results you write to the current directory are saved as output.
```

2. The second step is to read a CSV file named '20140711.csv' that contains the reprocessed data set into a Pandas DataFrame which allows for data manipulation and analysis in Python.



```
data = pd.read_csv('../input/unisys/ptsboardingsummary/20140711.CSV')
```

3. Some Important external data fields calculation

   IsHoliday Number of public holidays within that week

   DistanceFromCentre Distance measure from the city center

   For Calculating Distance between center with other bus stops by using Longitude and Latitude we have used the Haversine formula

```
from math import sin, cos, sqrt, atan2, radians
def calc_dist(lat1,lon1):
    ## approximate radius of earth in km
    R = 6373.0
    dlon = radians(138.604801) - radians(lon1)
    dlat = radians(-34.921247) - radians(lat1)
    a = sin(dlat / 2)**2 + cos(radians(lat1)) * cos(radians(-34.921247)) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    return R * c
```

```
'''Holidays--
2013-09-01,Father's Day
2013-10-07,Labour day
2013-12-25,Christmas day
2013-12-26,Proclamation Day
2014-01-01,New Year
2014-01-27,Australia Day
2014-03-10,March Public Holiday
2014-04-18,Good Friday
2014-04-19,Easter Saturday
2014-04-21,Easter Monday
2014-04-25,Anzac Day
2014-06-09,Queen's Birthday'''
```

4. Aggregate the Data According to Weeks and Stop names

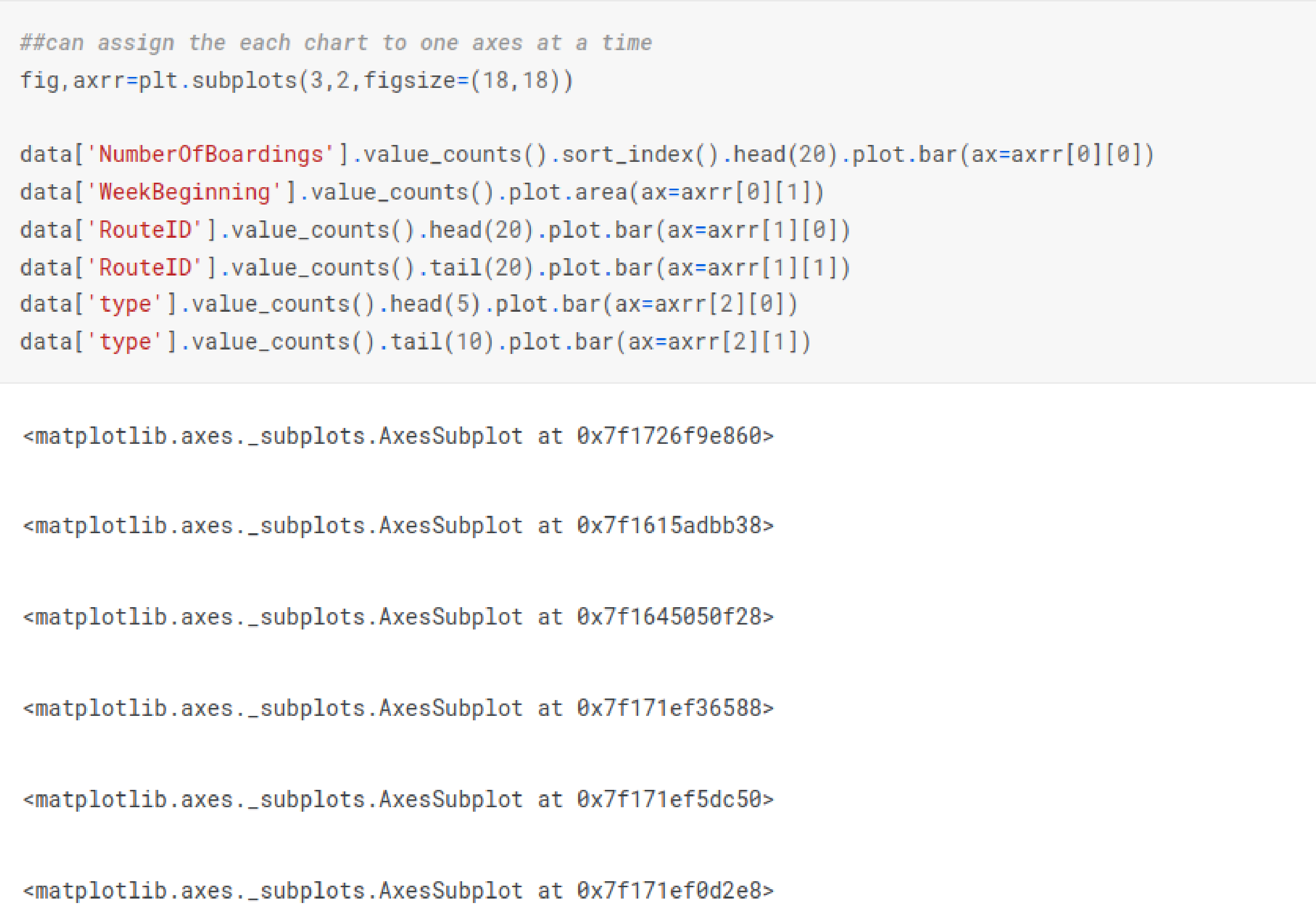
   NumberOfBoardings_sum Number of Boardings within particular week for each Bus stop

   NumberOfBoardings_count Number of times data is recorded within week

   NumberOfBoardings_max Maximum number of boarding done at single time within week

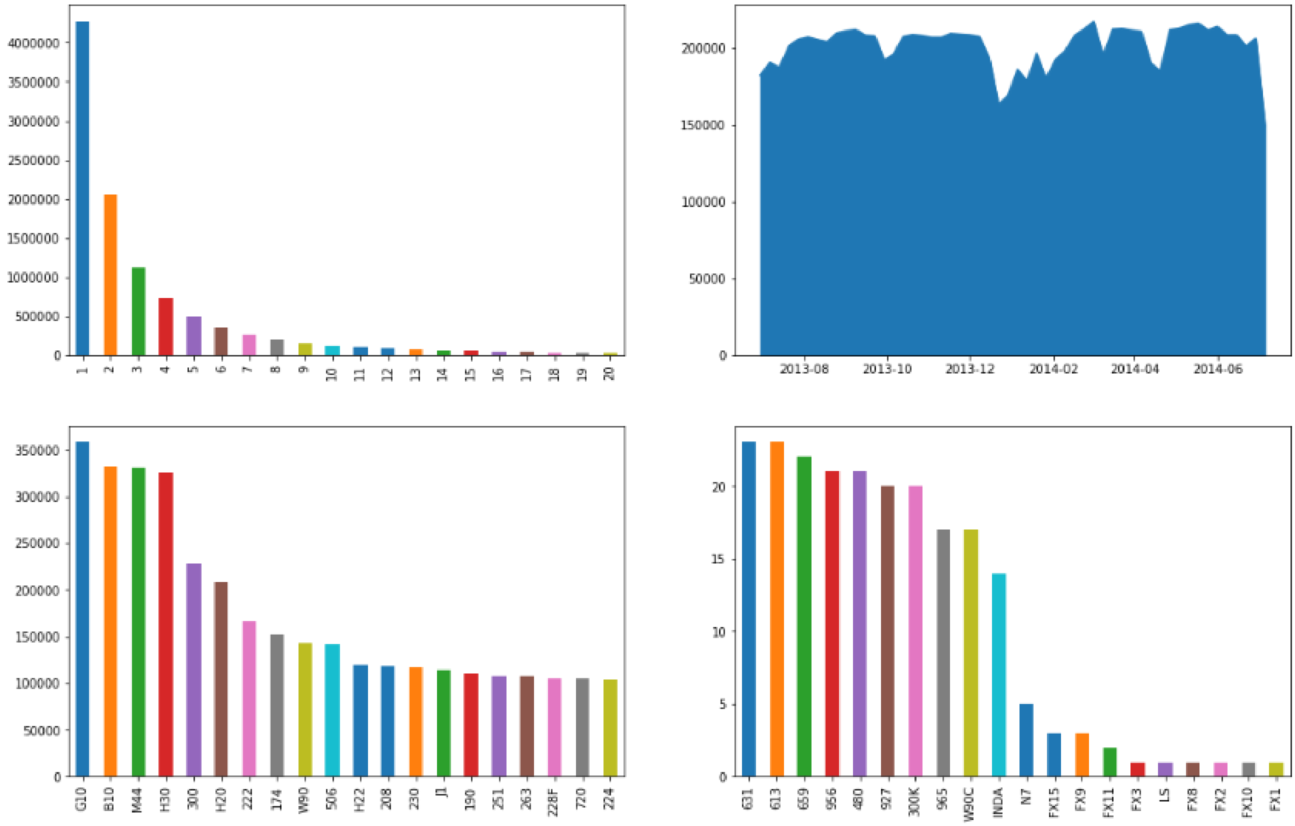
```
# st_week_grp1 = pd.DataFrame(data.groupby(['StopName','WeekBeginning','type']).agg({'NumberOfBoardings': ['sum', 'count']})).reset_index()
grouped = data.groupby(['StopName','WeekBeginning','type']).agg({'NumberOfBoardings': ['sum', 'count','max']})
grouped.columns = ["_".join(x) for x in grouped.columns.ravel()]
```

5. Next we plot the Number of Boarding and WeekBeginning and Route ID



```
##can assign the each chart to one axes at a time
fig,axrr=plt.subplots(3,2,figsize=(18,18))

data['NumberOfBoardings'].value_counts().sort_index().head(20).plot.bar(ax=axrr[0][0])
data['WeekBeginning'].value_counts().plot.area(ax=axrr[0][1])
data['RouteID'].value_counts().head(20).plot.bar(ax=axrr[1][0])
data['RouteID'].value_counts().tail(20).plot.bar(ax=axrr[1][1])
data['type'].value_counts().head(5).plot.bar(ax=axrr[2][0])
data['type'].value_counts().tail(10).plot.bar(ax=axrr[2][1])
```

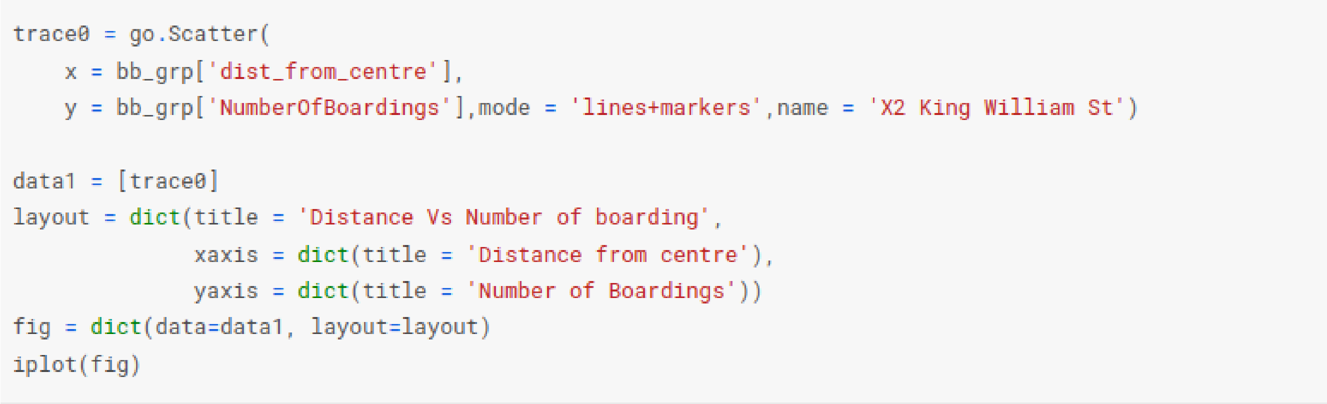
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1726f9e860>


<matplotlib.axes._subplots.AxesSubplot at 0x7f1615adbb38>


<matplotlib.axes._subplots.AxesSubplot at 0x7f1645050f28>


<matplotlib.axes._subplots.AxesSubplot at 0x7f171ef36588>


<matplotlib.axes._subplots.AxesSubplot at 0x7f171ef5dc50>


<matplotlib.axes._subplots.AxesSubplot at 0x7f171ef0d2e8>
```



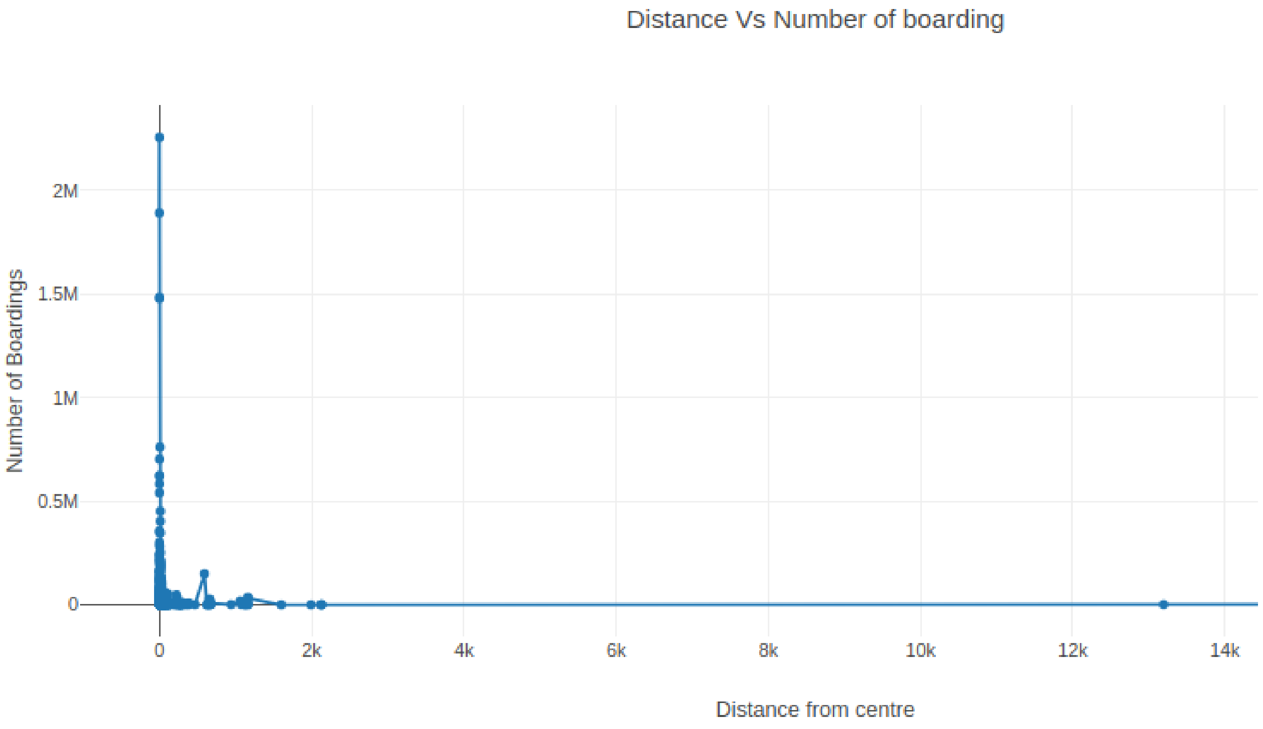
Inferences:

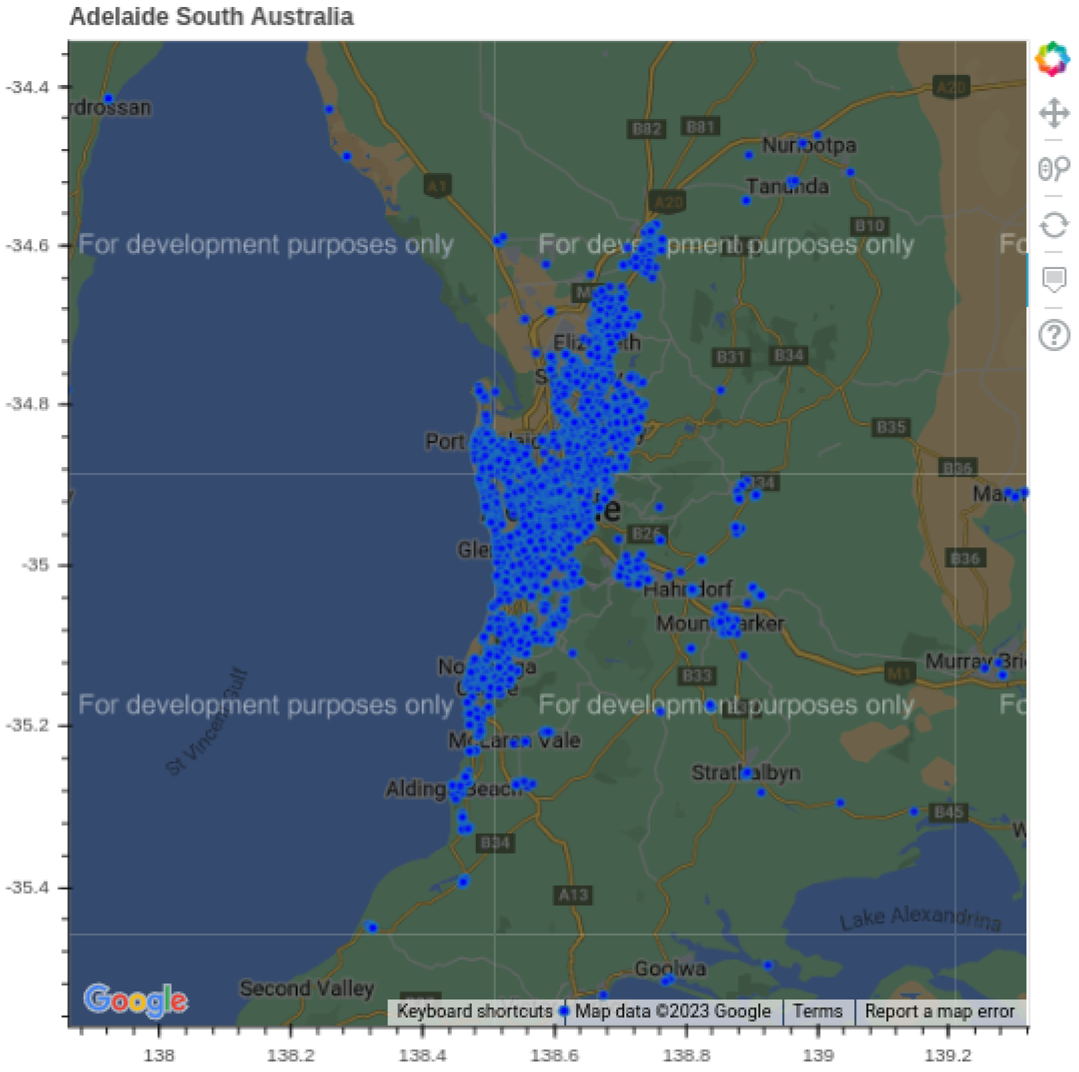
- More than 40 lakhs times only single people board from the bus stop.
- There are an average of 1.8 lakhs people who travel every week by bus in the Adelaide metropolitan area.
- G10,B10,M44,H30 are the most busiest routes in the city while FX8,FX3,FX10,FX1,FX2 are the least.
- Most of the Bus stops are Street_Address Type while there are very few which are store or post offices

6. Next we find the relation between the number of people boarding and the distance they travel



```
trace0 = go.Scatter(
    x = bb_grp['dist_from_centre'],
    y = bb_grp['NumberOfBoardings'],mode = 'lines+markers',name = 'X2 King William St')

data1 = [trace0]
layout = dict(title = 'Distance Vs Number of boarding',
              xaxis = dict(title = 'Distance from centre'),
              yaxis = dict(title = 'Number of Boardings'))
fig = dict(data=data1, layout=layout)
iplot(fig)
```

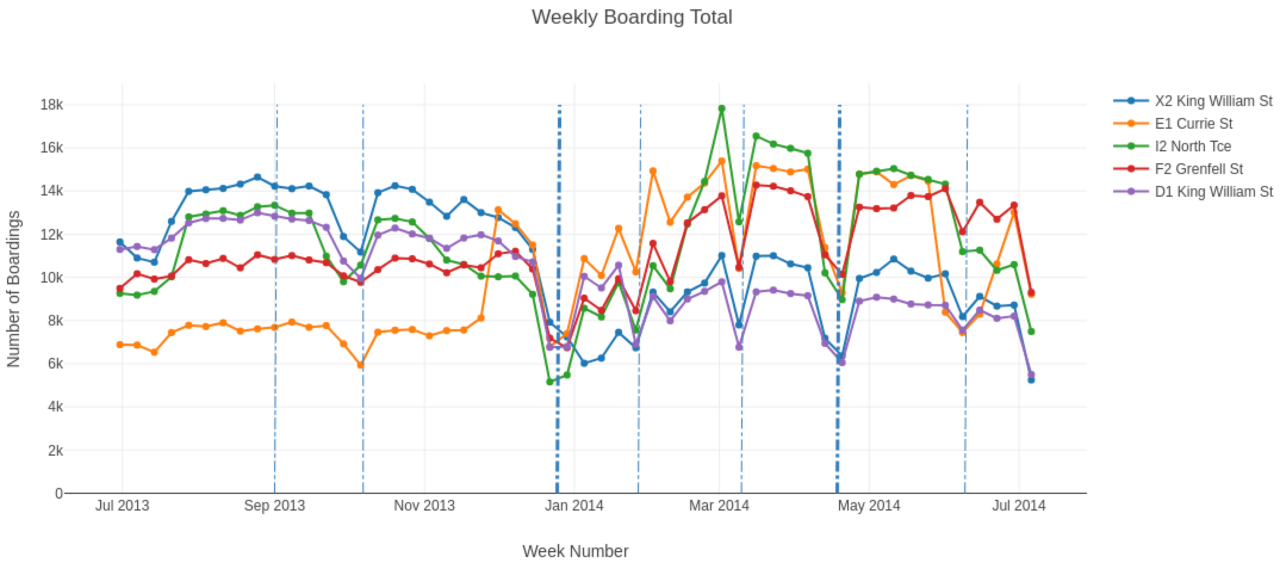




Inferences:

- As we move away from center the number of Boarding decreases
- There are cluster of bus stops near to the main Adelaide city as opposed to outside.so that's why most of boardings are near to center

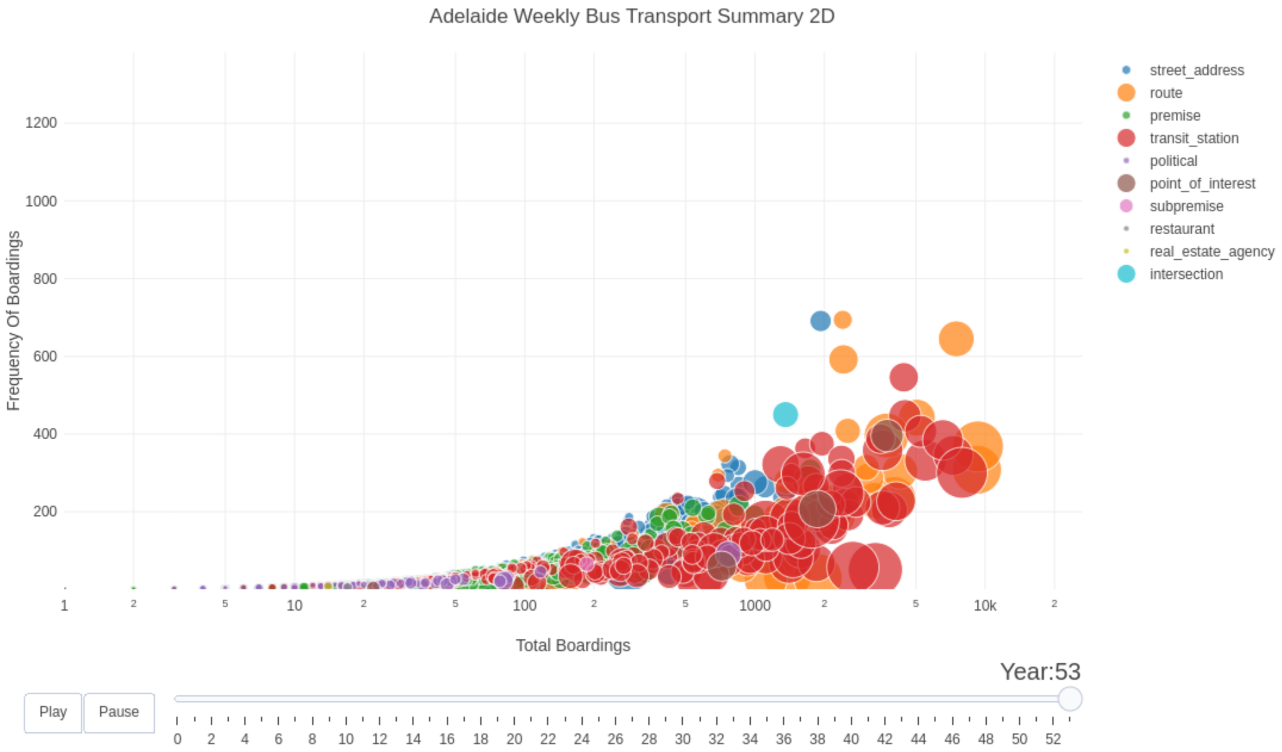
7. Now will plot the weekly basis boarding



Inferences:

- X2 King William St and stops near to that are the most busiest stops in the city. which have a number of boardings per week more than 10k.
- Vertical lines are the indicator of holidays which came within that week.
- Whenever there is any Public holiday that week period has less than average number of people traveling from bus.

8. Now we plot the Adelaide Weekly Bus transport Summary.

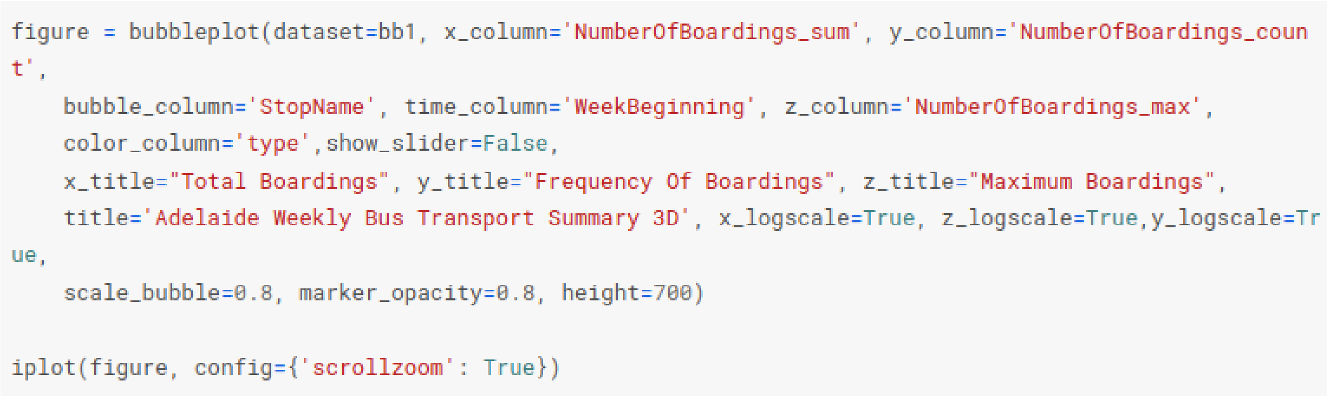


In the graph above, the size corresponds to the maximum number of people board at single time and the Total boardings and Frequency of boardings with stop name can be seen by hovering over the cursor on the bubbles.

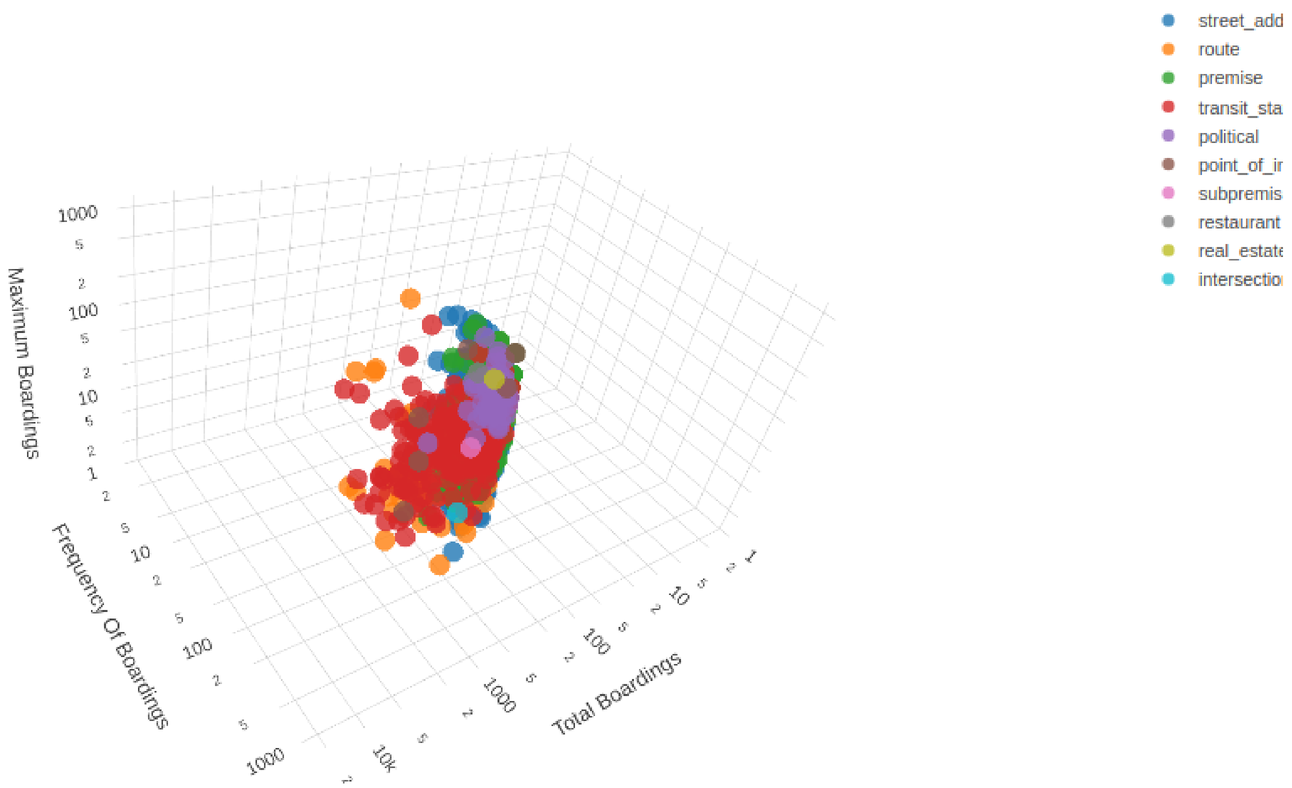
The animated bubble charts convey a great deal of information since they can accommodate upto seven variables in total, namely:

- X-axis (Total Boardings per week)
- Y-axis (Frequency of Bus Boarding)
- Bubbles (Bus stop name)
- Time (in week period)
- Size of bubbles (maximum number of people board at single time)
- Color of bubbles (Type of Bus stop)

9. Now we plot the Adelaide Weekly Bus transport Summary.



```
figure = bubbleplot(dataset=bb1, x_column='NumberOfBoardings_sum', y_column='NumberOfBoardings_coun
t',
    bubble_column='StopName', time_column='WeekBeginning', z_column='NumberOfBoardings_max',
    color_column='type',show_slider=False,
    x_title="Total Boardings", y_title="Frequency Of Boardings", z_title="Maximum Boardings",
    title='Adelaide Weekly Bus Transport Summary 3D', x_logscale=True, z_logscale=True,y_logscale=Tr
ue,
    scale_bubble=0.8, marker_opacity=0.8, height=700)

iplot(figure, config={'scrollzoom': True})
```



Inferences:

- Total Boardings are directly proportional to the frequency of bus boarding.
- In the 3D Plot we can see the cluster of address types.

Conclusion :

Thus the demographic analysis was performed for the Australian Traffic dataset using python libraries like pandas, numpy, seaborn and matplotlib.